Computer and Network Security

Lecture 9

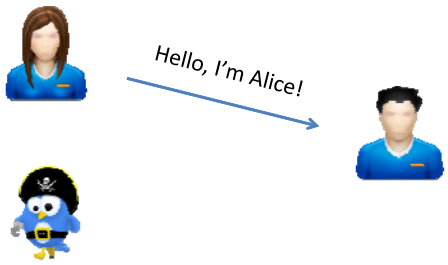
Authentication & Key Distribution

Outline

- Key Distribution Center
- Certification Authority
- Protocols & attacks

Authentication

- Bob Wants Alice to prove her identity to him
- 1st try: Alice says "Hello, I'm Alice!"

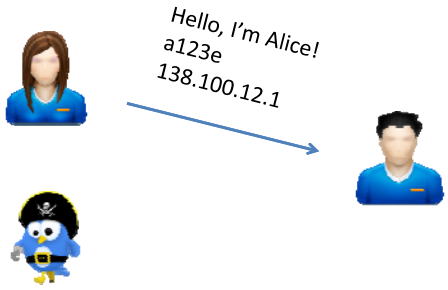


Authentication

- 2nd try: Alice says "Hello, I'm Alice!"
  - Within an IP packet containing her IP address

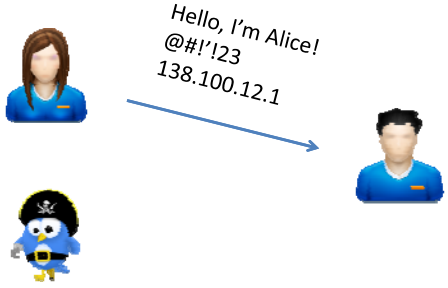
Authentication

- 3rd try: Alice says "Hello, I'm Alice!"
  - Within an IP packet containing her IP address
  - And her "secret" password



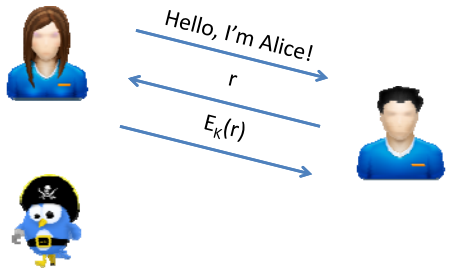
Authentication

- 4th try: Alice says "Hello, I'm Alice!"
  - Within an IP packet containing her IP address
  - And her "secret" password encrypted



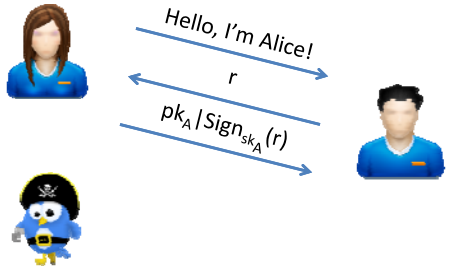
Authentication

- 5th try: Alice says "Hello, I'm Alice!"
  - Replies to a challenge using the key shared with Bob



Authentication

- 6th try: Can we use Public Keys?



Key Distribution

- Symmetric Crypto requires Alice and Bob to share a key
  - How to distribute the key securely?
- Asymmetric Crypto requires Alice and Bob to exchange their public keys
  - How to make sure that the right key is being used?

Trusted intermediaries

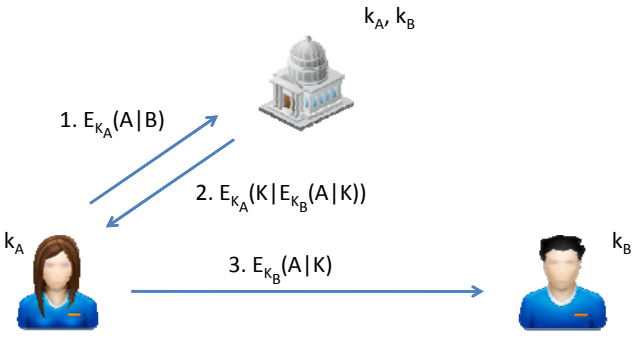
- Symmetric key problem:
  - How do two entities establish shared secret key over a distance?
- Solution:
  - Mutually trusted on-line key distribution center (KDC) acts as intermediary between entities
- Public key problem:
  - When Alice gets Bob's public key (from a web site, email), how does she know it is really Bob's?
- Solution:
  - Trusted off-line certification authority (CA)

Key Distribution Center (KDC)

- Responsible for distributing keys to pairs of users (hosts, processes, applications)
- Each user must share a unique key with the KDC
  - Use this key to communicate with
  - Each master key is in some off-line fashion

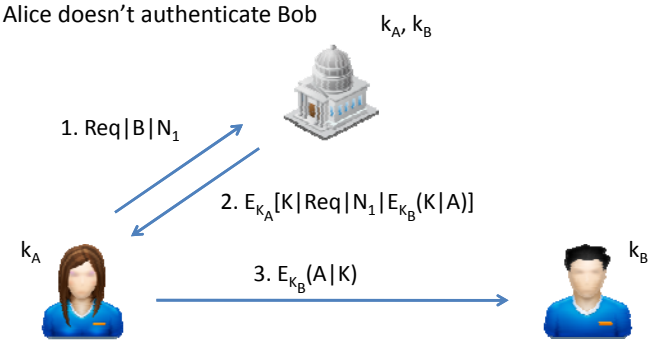
Key Distribution Center

- Remarks
  - Msg 2 is not tied to Msg 1
  - Any message is possibly old
  - Bob and Alice don't authenticate each other



Key Distribution Scenario

- Remarks
  - Msg 2 is tied to Msg 1
  - Msg 2 is fresh/new
  - Msg 1 and Msg 3 are possibly old
  - KDC doesn't authenticate Alice
  - Bob authenticates KDC
  - Alice doesn't authenticate Bob

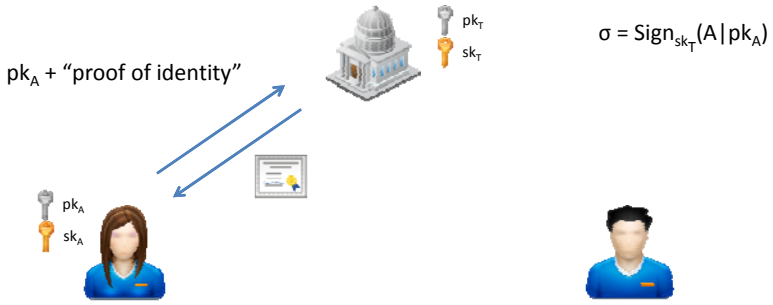


Public Key distribution

- Distribution of public key
  - Does not need to be secret
  - Need to be authentic
- Public announcement
  - E.g., in a newsgroup
  - Can be forged
- Public Key Certificates (PKCs)
  - Issued by Certification Authorities (CAs)
    - trusted
    - off-line

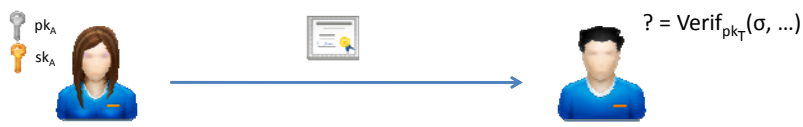
Certification Authority (CA)

- Binds public key to a specific entity
  - Alice registers its PK with CA
    - Provides "proof of identity" to CA
  - CA creates certificate binding Alice to this PK
    - signed with CA's secret key

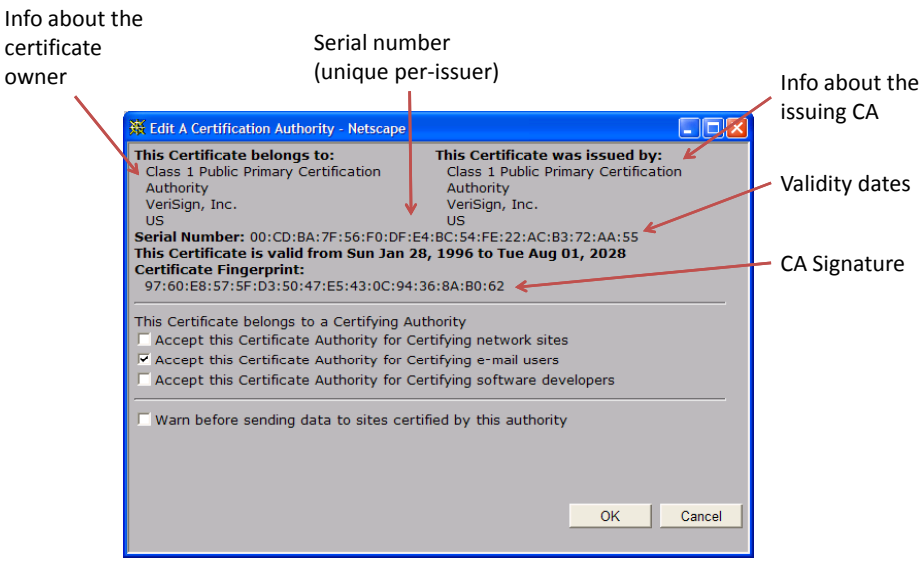


Certification Authority (CA)

- When Bob want to communicate with Alice
  - Get Alice's certificate (from her or elsewhere)
  - Check for expiration
  - Use CA's public key to verify the signature on Alice's certificate
  - Check for revocation (we'll talk about this later)
  - Extract Alice's public key

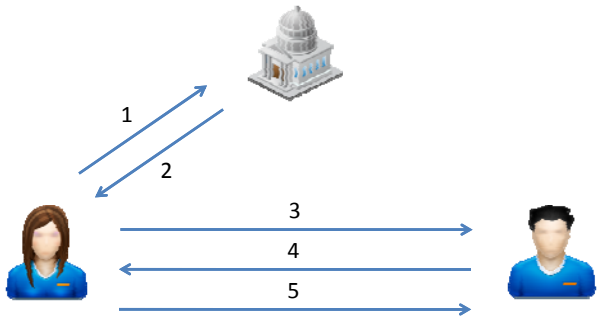


What's in a Certificate?



Needham-Schroeder Protocol (1978): 1st distributed security protocol

1. $A \rightarrow T$: $A|B|N_A$
2. $T \rightarrow A$: $E_{k_A}(N_A|B|K|E_{k_B}(A|K))$
3. $A \rightarrow B$: $E_{k_B}(A|K)$
4. $B \rightarrow A$: $E_K(N_B)$
5. $A \rightarrow B$: $E_K(N_B-1)$



Security?

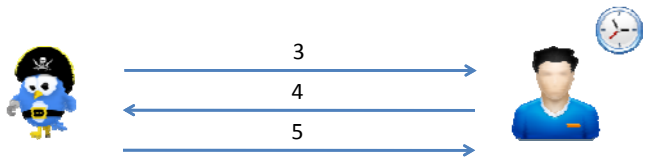
- Denning-Sacco Attack
  - Eve recorded an old session for which session K is known to her

1. $A \rightarrow T: A|B|N_A$
2. $T \rightarrow A: E_{k_A}(N_A|B|K|E_{k_B}(A|K))$
3. $A \rightarrow B: E_{k_B}(A|K)$
4. $B \rightarrow A: E_K(N_B)$
5. $A \rightarrow B: E_K(N_B\text{-}1)$

---

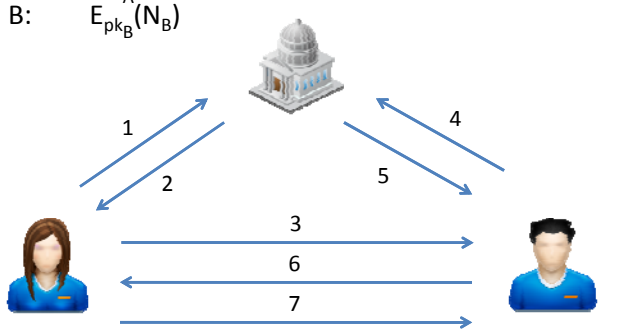
1. $E \rightarrow B: E_{k_B}(A|K)$
2. $B \rightarrow E: E_K(N_{B'})$
3. $A \rightarrow B: E_K(N_{B'}\text{-}1)$

- No freshness guarantee for message 3
- Can be fixed by adding timestamps



PK-based Needham-Schroeder Protocol

1. $A \rightarrow T: A|B$
2. $T \rightarrow A: Sign_{sk_T}(PK_B|B)$
3. $A \rightarrow B: E_{pk_B}(N_A|A)$
4. $B \rightarrow T: B|A$
5. $T \rightarrow B: Sign_{sk_T}(PK_A|A)$
6. $B \rightarrow A: E_{pk_A}(N_A|N_B)$
7. $A \rightarrow B: E_{pk_B}(N_B)$

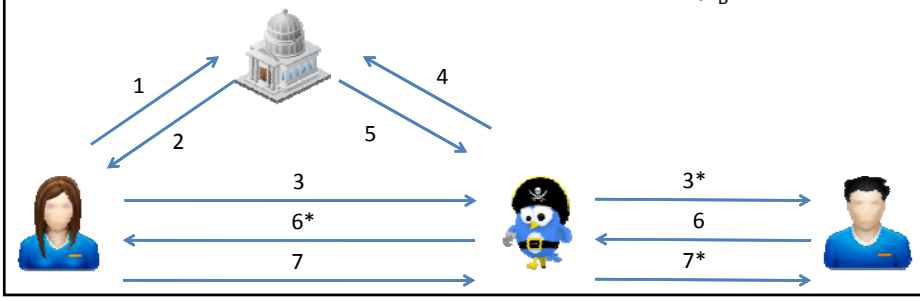


Security?

- PK delivery messages
  - Do not guarantee freshness of the public keys
  - How to solve it?
    - Timestamp in messages 2 and 5 or challenges in messages 1&2 and 4&5
    - Public Key Certificate:
      - assign expiration time/data to each certificate

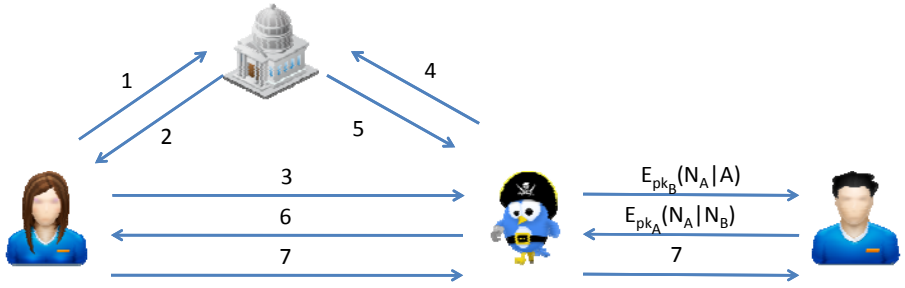
Impersonation by Interleaving (Lowe's attack)

1. $A \rightarrow T$: $A \mid B$
2. $T \rightarrow A$: $Sign_{sk_T}(PK_B \mid B)$
3. $A \rightarrow B$: $E_{pk_B}(N_A \mid A)$
4. $B \rightarrow T$: $B \mid A$
5. $T \rightarrow B$: $Sign_{sk_T}(PK_A \mid A)$
6. $B \rightarrow A$: $E_{pk_A}(N_A \mid N_B)$
7. $A \rightarrow B$: $E_{pk_B}(N_B)$



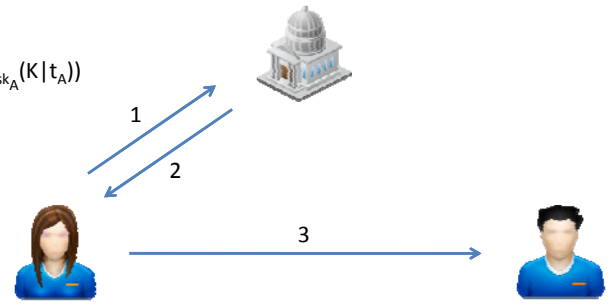
Impersonation by Interleaving (Lowe's attack)

3. A → E: $E_{pk_E}(N_A|A)$
4. B → T: $B|A$
5. T → B: $Sign_{sk_T}(PK_A|A)$
6. B → A: $E_{pk_A}(N_A|N_B)$
7. A → B: $E_{pk_B}(N_B)$



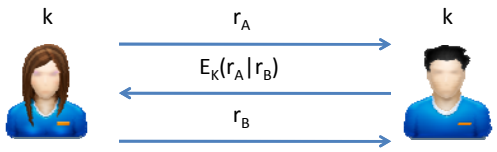
PK-based Denning Sacco Attack

1. A → T: A|B
2. T → A: $Cert_A|Cert_B$
3. A → B: $Cert_A|E_{pk_B}(K|t_A|Sign_{sk_A}(K|t_A))$



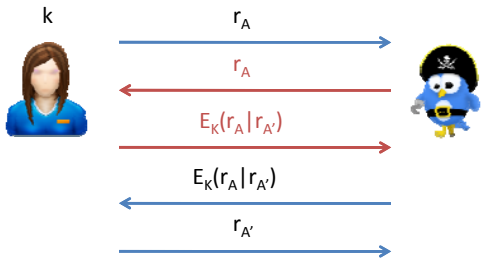
4. B → C: $Cert_A|E_{pk_C}(K|Sign_{sk_A}(K|t_A))$

Reflection Attack



- Authentication through shared secret
  - Only who knows k can
    - Encrypt r_A
    - Decrypt r_B

Reflection Attack



- Fix
  - Use two different keys k_{AB} and k_{BA}
  - Remove simmerty
    - Msg identifiers, sender/receiver

Lessons learned?

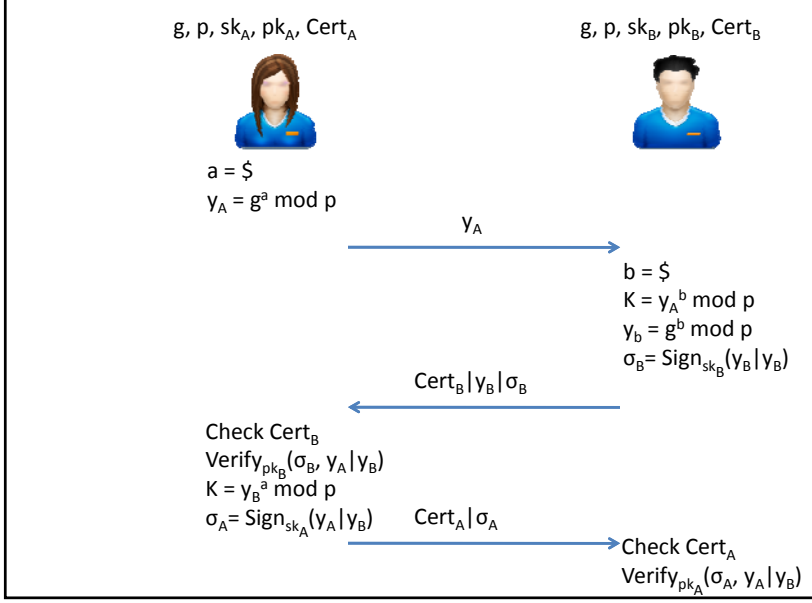
- Designing **secure** protocols is hard
- **Many** documented failures in the literature
- Good protocols are already standardized
  - e.g., ISO 9798, X.509, … – use them!
- Verifying security gets much harder as protocols get more complex
  - more parties, messages round

27

Hints for a secure protocol

- Break symmetry
  - Identifiers
    - Message
    - Sender/Receiver
  - Nonces
    - Freshness
    - Must be unpredictable
  - Timestamp
    - Timeliness
    - Require clock synchronization
  - Counters
    - Stateful

Station-to-Station



X.509 Authentication and Key Distribution

$\sigma_A = Sign_{sk_A}(1|t_A|r_A|B|other_A|E_{pk_B}(K_{AB}))$

$1|t_A|r_A|B|other_A|E_{pk_B}(K_{AB})|\ \sigma_A$ →

$\sigma_A = Sign_{sk_A}(2|t_A|r_A|B|other_A|E_{pk_B}(K_{AB}))$

$2|t_A|r_A|B|other_A|E_{pk_B}(K_{AB})|\ \sigma_A$ →

$\sigma_B = Sign_{sk_B}(2|t_B|r_B|A|r_A|other_B|E_{pk_B}(K_{BA}))$

$2|t_B|r_B|A|r_A|other_B|E_{pk_B}(K_{BA})|\ \sigma_B$ ←

$3|t_A|r_A|B|other_A|E_{pk_B}(K_{AB})|\ \sigma_A$ →

$3|t_B|r_B|A|r_A|other_B|E_{pk_B}(K_{BA})|\ \sigma_B$ ←

$\sigma_A = Sign_{sk_A}(3|r_B)$

$3|r_B|\sigma_A$ →